Popular Computing

BOX 272
CALABASAS
CA., 91302

Vol 1 No 3 June 1973

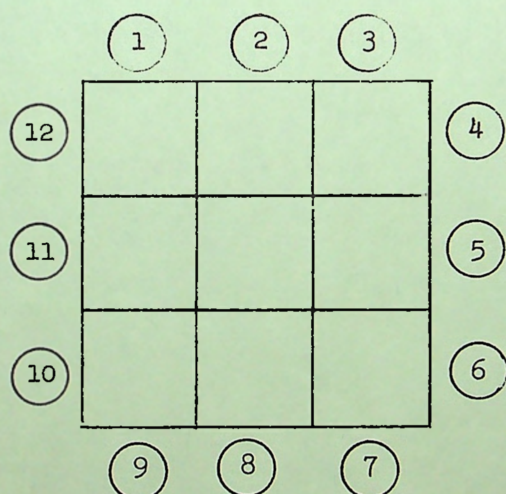
Fourway

The Fourway problem was devised to illustrate two principles:

1. That a simple, well-defined procedure cannot be carried out manually, but requires a computer.

2. That such a procedure, programmed by a precise algorithm, is unpredictable in the sense that the person who writes the program cannot tell what will happen when it runs on the computer.

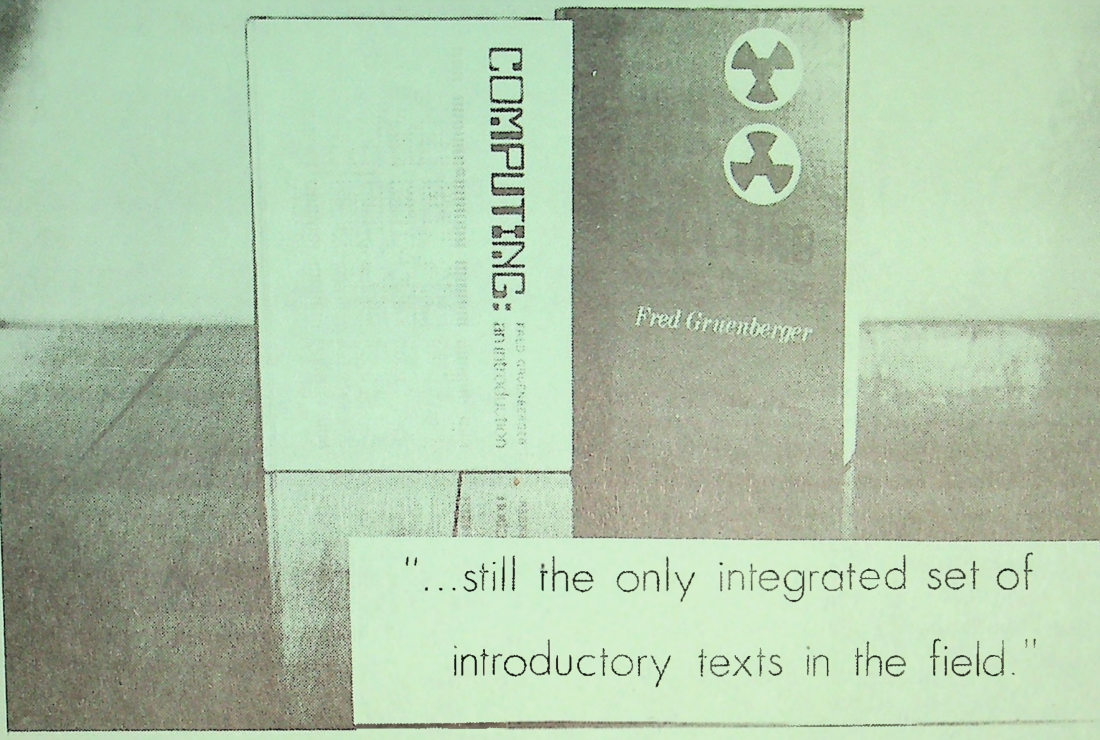
Consider the 3 x 3 form of Fourway. Given an array of nine cells:



Please turn to page 3

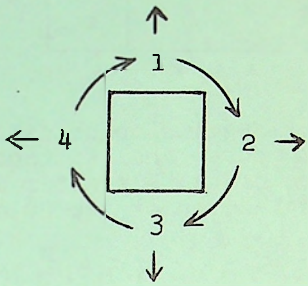
COMPUTING: AN INTRODUCTION, Harcourt Brace Jovanovich, 1969
757 Third Avenue, New York City 10017

COMPUTING: A SECOND COURSE, Canfield Press, 1971
850 Montgomery Street, San Francisco 94133



POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91301. Subscription rate in the United States is $15 per year, or $12 if remittance accompanies the order. For all foreign subscriptions, add $4.50 per year. Multiple subscriptions to the same address, add $5 each; thus, 3 copies per month is $25 per year, U.S. delivery. Back issues $1.50 each. Subscriptions may begin with any issue. Subscriptions for qualified undergraduate students half price.

Each cell contains an integer in the range from 1 to 4. The number in the cell indicates the direction to be followed, according to this plan:



Thus, if the cell contains a 1, move North; if it contains a 2, move East; and so on. Each time a move is made out of a cell, the number in the cell advances as shown by the arrows in the circle; that is, 1 advances to 2; 4 advances to 1; and so on.

The nine cells all contain 1 to start. Play starts at the center cell. Moves are made from cell to cell until an escape from the array occurs, to one of the 12 exits numbered in the first diagram. A tally is made of the exit number, and a new play begins.

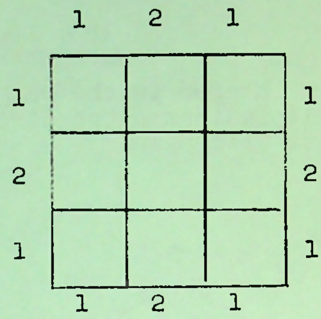
We now have the following problems:

1. Will each game, starting always at the center cell, eventually exit?

2. What will be the distribution of the tallies at the 12 exits after many games?

3. Will the pattern of numbers within the cells ever return to all 1's?

The 3 x 3 form can be analyzed completely by hand. The first play exits at 2; the second play exits at 3; the third play exits at 1. After 16 plays, the array will have returned to its original pattern, and the distribution of the exit tallies will be:

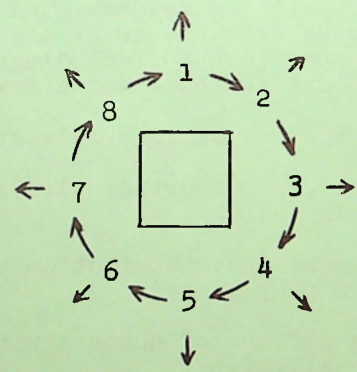


The 5 x 5 form begins to be difficult to analyze by hand. Computer analysis shows that it will return to the pattern of all 1's after 104 games, after which the distribution of the 20 exit tallies will be:

3,6,8,6,3,3,6,8,6,3,3,6,8,6,3,3,6,8,6,3.

Similar results for the 7 x 7 form indicate a cycle length of 544, and for the 9 x 9 form, a cycle length of 146248. Results are not known for any larger form of Fourway.

The next higher extension is to Eightway, in which each cell can contain a number from 1 to 8, with the following move rule:



The cycle lengths for Eightway have been found as follows:

3 x 3: 140

5 x 5: 69784

and nothing further is known.

If the problem is extended to three dimensions, then the simplest form would be Sixway, in which a move can be made from one cube to any of the six cubes surrounding it, proceeding across the six faces of the cube. If movement across the edges is allowed, then the game becomes 18-way; or if movement to <u>any</u> surrounding cube is allowed, the game becomes 26-way. For the 5 x 5 x 5 form of 26-way, there are 218 possible exit points.

Fourway has been studied to some depth. There seems to be no way that a pattern of numbers in the cells can cause a game to hang up in a loop, although this has not been proven. For the smaller forms (e.g., 5 x 5), every starting pattern that has been tried has yielded the same results; namely, a return to that pattern after the stated cycle length.

3

Log 3	0.47712125471966243729502790325511530920012886419070
Ln 3	1.09861228866810969139524523692252570464749055782275
$\sqrt{3}$	1.73205080756887729352744634150587236694280525381038
$\sqrt[3]{3}$	1.44224957030740838232163831078010958839186925349935
$\sqrt[5]{3}$	1.24573093961551732596668033664030508093930999306878
$\sqrt[7]{3}$	1.16993081275868688646297572551373466769940419642093
$\sqrt[10]{3}$	1.11612317403390443444261413837709258111831692535068
$\sqrt[100]{3}$	1.01104669193785359065566004544576737820871882795664
e^3	20.085536923187667740928529654581717896987907838554 2
π^3	31.006276680299820175476315067101395202225288565885 1
$\tan^{-1} 3$	1.24904577239825442582991707728109012307782940412990
3^{100}	515377520732011331036461129765621272702107522001.
3^{1000}	1322070819480806636890455259752144365965422032752148167664920368226828597346704899540778313850608061963909777696872582355950954582100618911865342725257953674027620225198320803878014774228964841274390400117588618041128947815623094438061566173054086674490506178125480344405547054397038895817465368254916136220830268563778582290228416398307887896918556404084898937609373242171846359938695516765018940588109060426089671438864102814350385648747165832010614366132173102768902855220001

N-SERIES

Desk Calculator Review

Texas Instruments SR-10

The SR-10 is the intermediate model in pocket electronic calculators, selling currently for $150. It has most of the features of the low priced models (those reviewed in PC-2), plus these features:

1. In addition to full floating decimal capability, there is also scientific notation, with a range on the exponent from +99 to -99.

2. Function keys are provided for reciprocals, squares, and square roots.

3. The battery-saving feature turns off the display, except for the low order digit, after 25 seconds or so of non-use; the complete display is restored on depressing the EQUALS key.

On the other hand, the SR-10 does not have the ability to store a constant multiplier or divisor. Also, the lower priced machines are capable of squaring any result (by depressing TIMES and EQUALS) and can calculate the reciprocal of a result, albeit awkwardly.

Texas Instruments has published an "Applications Guide" for the SR-10, showing many efficient tricks of desk calculator use, plus ways of obtaining logarithms, exponentials, and trigonometric functions. The latter schemes are sort of Hastings' approximations* but designed for a more limited range, limited accuracy, and for use on this machine. As an example of the former schemes, the booklet (20 pages) shows that sums of quotients can be calculated by identities like:

$$\frac{A}{B} + \frac{C}{D} + \frac{E}{F} = \left[\left(\frac{AD}{B} + C \right) \cdot \frac{F}{D} + E \right] \Big/ F$$

Thus, the SR-10 offers a wider arithmetic range through scientific notation; it has the square root function; it lacks storage for a constant; the maker is interested in helping users achieve greater efficiency.

---

*Cecil Hastings, Jr., Approximations for Digital Computers, Princeton University Press, 1955.

Numbering the Fractions

The accompanying table is part of an infinite array. It is a device for displaying all the proper fractions in lowest terms in order. Each row is a separate higher denominator, and within each row the numerators are in ascending sequence. Since things are neat and orderly, every such fraction will appear in the array in a definite place, and hence every fraction can be given a position number. The position numbers shown in circles on the left are for the fractions having one as a numerator. Every row will have such a fraction, and will also have the fraction of the form (D-1)/D. This scheme for numbering the fractions is a thinly disguised way of expressing the Euler ϕ-function.

The present limits of knowledge about this numbering scheme are summarized in this table:

Denominator	Number of unit fraction
100	3005
200	12153
300	27319
400	48519
500	75917
600	109341
700	148779
800	194431
900	246087
1000	303793
2000	1215789
3000	2735389
4000	4862003
5000	7598459
6000	10941565
7000	14892747
8000	19452583
9000	24619119
10000	30393487
12500	47493359
15000	68390317
20000	121582397
25000	189970091
28000	238299681

These results are due to Richard Sandin, April 15, 1972

1	1/1											
2	1/2											
3	1/3	2/3										
5	1/4	3/4										
7	1/5	2/5	3/5	4/5								
11	1/6	5/6										
13	1/7	2/7	3/7	4/7	5/7	6/7						
19	1/8	3/8	5/8	7/8								
23	1/9	2/9	4/9	5/9	7/9	8/9						
29	1/10	3/10	7/10	9/10								
33	1/11	2/11	3/11	4/11	5/11	6/11	7/11	8/11	9/11	10/11		
43	1/12	5/12	7/12	11/12								
47	1/13	2/13	3/13	4/13	5/13	6/13	7/13	8/13	9/13	10/13	11/13	12/13
59	1/14	3/14	5/14	9/14	11/14	13/14						
65	1/15	2/15	4/15	7/15	8/15	11/15	13/15	14/15				
73	1/16	3/16	5/16	7/16	9/16	11/16	13/16	15/16				
81	1/17	2/17	3/17	4/17	5/17	6/17	7/17	8/17	.......15/17	16/17		
97	1/18	5/18	7/18	11/18	13/18	17/18						
103	1/19	2/19	3/19	4/19	5/19	6/19	7/19	8/19	..........17/19	18/19		

Systematic array of the proper fractions in lowest terms. The position number of the fractions with unit numerators is indicated with the circled figures.

Figure 1 shows the basic logic for extending the numbered array indefinitely. The housekeeping phase might be the following: Set N = 1, D = 25000, and P = 189970091.

Whatever we might want to do with the array can be added at the place marked X. Questions like the following could be answered:

a) What is the position number of the fraction 2345/10007?

b) What fraction has the position number 500,000,000?

The logic of the first flowchart may be correct. It constitutes an algorithm, and illustrates clearly that an algorithm, while guaranteeing results, may be terribly inefficient. Casual inspection of Figure 1 reveals that there may be many ways to speed up the calculation:

1. Since every row of the array contains both 1/D and (D-1)/D, it should be possible to by-pass the test logic for those fractions.

2. If the denominator is even, then only odd numerators need be considered.

3. If the denominator is prime, its whole row can be by-passed, and the position number can simply be increased by (D-1). This introduces a nice matter of judgement; namely, will a test for primality cost more than it saves? Note also that if shortcut (1) has been added to the logic, then <u>this</u> shortcut must allow for it.

4. Since there are always an even number of fractions in any row, then we need only count to the middle of the row and double that count. For the cost of one simple test, the speed of the algorithm can be doubled.

If all of these shortcuts could be applied at once, the scheme would operate nearly three times as fast. A factor of 3 increase in operating speed is worth working for.

Figure 2 shows an improved scheme for extending the array. The whole problem makes an excellent training exercise in the logic of flowcharting.

Housekeeping: Assign values to N (Numerator), D (Denominator), and P (Position number).

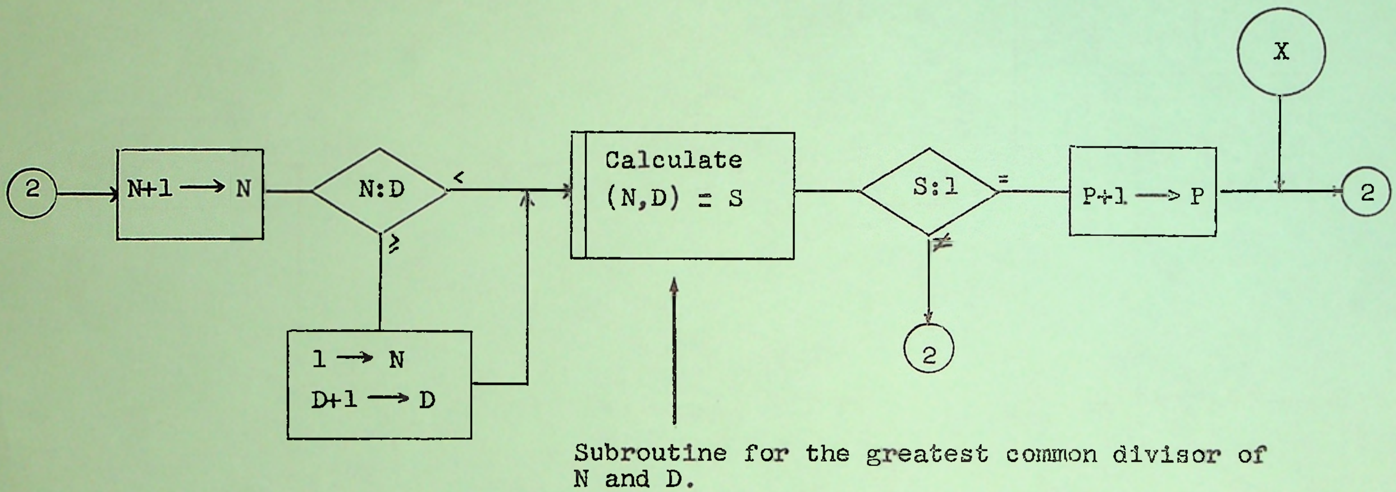


Figure 1. The basic logic for extending the array.

Housekeeping: Set values for N (Numerator), D (Denominator), and P (Position number). Set K = 1 (always safe).

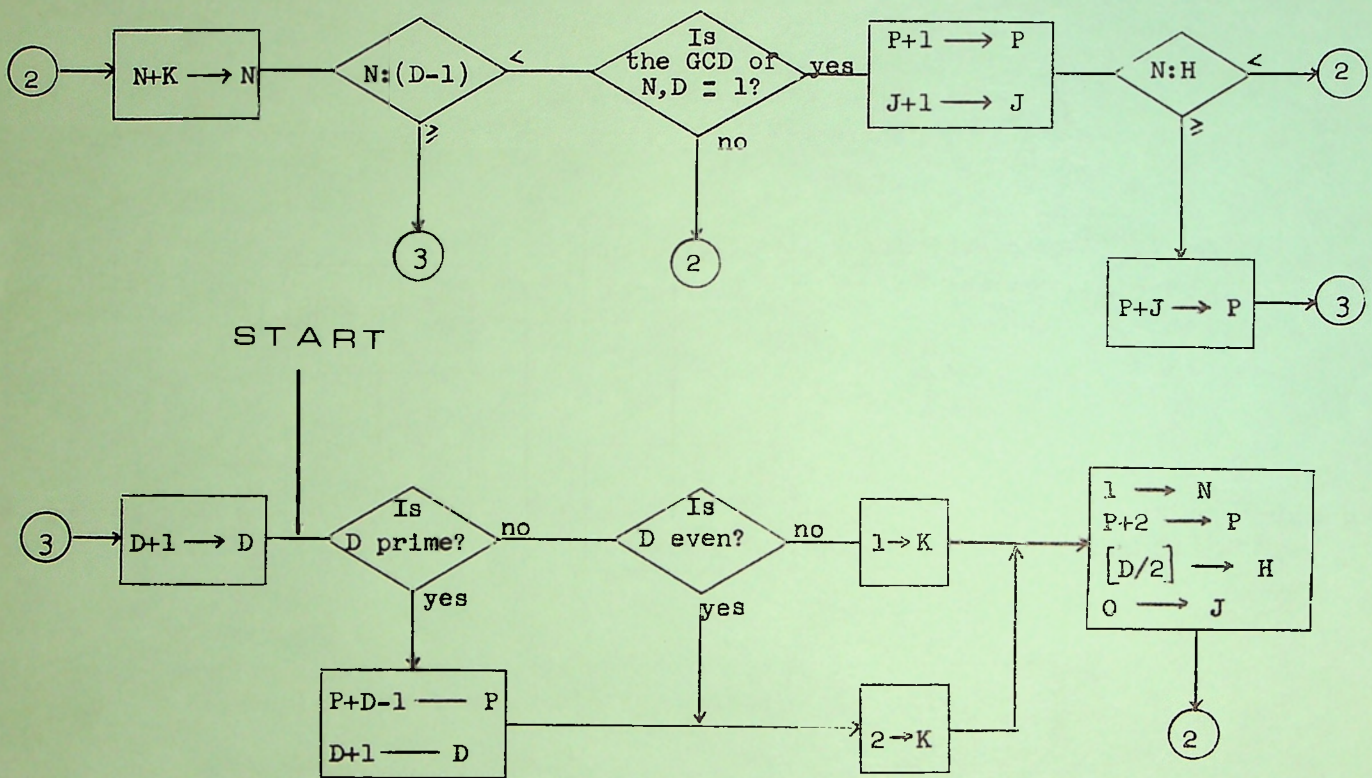


Figure 2. An improved scheme for numbering the fractions. H is the length of half a row. J maintains the count of the number of fractions in half a row.

The A^B Problem

It is known that when $x \approx \pi\sqrt{163}$, then

$e^x \approx 262537412640768743.99999999999925007260\ldots$

This is probably the closest value to an integer known for an irrational number. The value of A^B, where A and B are irrationals, is always irrational, if not transcendental. The table on the next page shows some results for A and B taken as square roots of small primes; for example, $\sqrt{19}^{\sqrt{11}} \approx 131.997009$. These results were obtained with an HP-35 calculator.

If A and B are restricted to square roots of non-square integers (not necessarily primes), how close can we get to an integer? In the example just given, we have come to within .0002991. Call this G. If we let $\sqrt{A}^{\sqrt{B}} = Q$, then we seek the smallest value of

$$[Q + 1] - Q$$

$$\text{or} \qquad Q - [Q]$$

where the brackets denote "greatest integer in."

A year's subscription to POPULAR COMPUTING will be given to the person finding the smallest value of G, subject to the restrictions stated above. Entries received up to August 1, 1973 will be considered. Evidence must be submitted that the computer program was properly tested.

B

A	2	3	5	7	11	13
2	1.632526919	1.822634654	2.170509877	2.501642535	3.156470775	3.488908194
3	2.174581429	2.589399904	3.415370162	4.277323451	6.183239815	7.246738242
5	3.120659822	4.030192405	6.046056782	8.407181096	14.42482727	18.20061633
7	3.958900210	5.393570645	8.807412399	13.12077909	25.20208981	33.38275365
11	5.449740577	7.977604848	14.59864060	23.85794487	53.32788958	75.40451682
13	6.133055549	9.219412396	17.59650954	29.75832911	70.35031865	101.9036964
17	7.414105749	11.63054388	23.75109928	42.43565633	109.7662172	165.2811547
19	8.020756700	12.80657815	26.89614202	49.16227917	131.9997009	201.9788238
23	9.180936567	15.11091975	33.30105586	63.29896422	181.2049884	285.0294716
29	10.81612109	18.47029437	43.15296751	86.01431002	266.1415774	432.8879364
31	11.33840502	19.56848093	46.49358797	93.94768390	297.2648657	488.1918535
37	12.84949539	22.80880496	56.66342580	118.7232792	398.6285102	671.6080027
41	13.81689322	24.92939798	63.55462606	135.9917308	472.6062489	808.1415663
43	14.29014455	25.97916446	67.03062249	144.8357105	511.4474137	880.5968683
47	15.21779516	28.05945444	74.03928865	162.9211812	592.7350428	1033.754188

The A^B Problem